

# **IB DISCUSS**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &  
ENGINEERING**

BY

**Yusra Zaidi**

**EN16CS301299**

Under the Guidance of

**Prof. Sachin Solanki**

**Miss Ekta Goel**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**May 2020**

# **IB DISCUSS**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &  
ENGINEERING**

BY

**Yusra Zaidi  
EN16CS301299**

Under the Guidance of  
**Prof. Sachin Solanki  
Miss Ekta Goel**



**Department of Computer Science & Engineering  
Faculty of Engineering  
MEDI-CAPS UNIVERSITY, INDORE- 453331**

**May 2020**

## **Report Approval**

The project work “**IB DISCUSS**” is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation

Affiliation

External Examiner

Name:

Designation

Affiliation

## **Declaration**

I hereby declare that the project entitled “**IB DISCUSS**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science & Engineering’ completed under the supervision of **Prof. Sachin Solanki, Department of Computer Science & Engineering**, Faculty of Engineering, Medi-Caps University, Indore and **Miss Ekta Goel, Mentor**, InterviewBit Software Services LLP is an authentic work.

Further, I declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Yusra Zaidi**

**EN16CS301299**

## **Certificate**

We, **Prof. Sachin Solanki** and **Miss Ekta Goel** certify that the project entitled **IB DISCUSS** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science & Engineering by **YUSRA ZAIDI**, EN16CS301299 is the record carried out by her under our guidance and that the work has not formed the basis of award of any other degree elsewhere.

---

**Prof. Sachin Solanki**

Computer Science & Engineering

Medi-Caps University, Indore

---

**Miss Ekta Goel**

Mentor

InterviewBit software services LLP

---

**Dr. Suresh Jain**

**Head of the Department**

Computer Science & Engineering

Medi-Caps University, Indore

## **Acknowledgements**

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Sunil K Somani**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda**, Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Suresh Jain** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **External Guide, Miss Ekta Goel**, Mentor, InterviewBit software services LLP as well as to my **Internal Guide, Prof. Sachin Solanki**, Department of Computer Science & Engineering, MU, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

**Yusra Zaidi**

EN16CS301299

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

## **Abstract**

IB Discuss aims to bring different compassionate coders on a common platform of the interview bit where they can post the blogs and can get help from the community. IB Discuss aims to communicate the thoughts of different users on a particular where other coders which are beginners or the coders which are struck in a problem or concept have a way out. It also provides skill coders to increase the strength of debugging a concept or problem which they have faced before they can help other users in a comment section of a particular blog provided which all the users can view. Guest users can also interact with platform so they can get an idea of how it works but these guest users are not allowed to add a comment or a blog on the IB Discuss.

## **Table of Contents**

		<b>Page No.</b>
	Report Approval	Ii
	Declaration	Iii
	Certificate	Iv
	Acknowledgement	V
	Abstract	Vi
	Table of Contents	vii
	List of figures	Ix
Chapter 1	Introduction	1
	1.1 Introduction	1
	1.2 Objectives	1
	1.3 Significance/Scope	2
Chapter 2	System Requirement Analysis	4
	2.1 Information Gathering	4
	2.2 System Feasibility	4
	2.2.1 Economical	4
	2.2.2 Technical	5
	2.2.3 Behavioral	5
	2.3 Platform Specification	5
	2.3.1 Hardware	5
	2.3.2 Software Implementation Technology	5
Chapter 3	Experimental Setup	7
	3.1 Landing Page	7
	3.2 Authorization of user	8
	3.3 Post a blog	8
	3.4 Editing and Deleting a post	9
	3.5 View Blogs and Comments	9
	3.6 Adding a comment	10
	3.7 Editing and deleting a comment	10
Chapter 4	System Analysis	11
	4.1 Information Flow Representation	11
	4.1.1 Activity Diagram	11



	4.2 Use case Diagram	12
	4.3 ER Diagram	15
Chapter 5	Testing	17
	5.1 Testing Objective	17
	5.2 Testing Scope	17
	5.3 Testing Mehods	17
	5.4 Test Cases	18
	5.5 Sample Test Data & Results	19
Chapter 6	Learning in Training	20
	6.1 NodeJS	20
	6.2 ExpressJS	20
	6.3 NPM	21
	6.4 Mongoose	22
	6.4.1 Collections	22
	6.4.2 Documents	22
	6.4.3 Fields	23
	6.4.4 Schema	23
	6.4.5 Models	23
	6.5 Data Structures and Algorithm	23
	6.5.1 Graph Algorithms	23
	6.5.2 Trie	23
	6.5.3 Segment Trees	24
	6.6 Working with Amazon EC2 Instance	24
	6.6.1 Features of Amazon EC2	25
Chapter 7	Summary and Conclusions	26
Chapter 8	Results and Discussions	27
Chapter 9	References	28

## List of Figures

Figure No.	Figure Name	Page No.
3.1	Landing Page	7
3.2	Login Page	8
3.3	New Blog	8
3.4	Edit/Delete Blog	9
3.5	View Blogs and Comments	9
3.6	Adding a comment	10
3.7	Edit/Delete Comment	10
4.1	Activity Diagram	12
4.2	Use Case Diagram	14
4.3	ER Diagram	16

# **CHAPTER 1.**

## **INTRODUCTION**

### **1.1 Introduction**

People are rapidly moving towards a smarter world, with implementation of smart cities, smart classrooms and smart phones and for that we need passionate coders which can code any problem according to the given scenario or given constraints. At present, information and communication technology has been brought to a number of business models in order to make the operation more convenient and effective. IB Discuss provides various emerging UI platform for the different coders available on the interview bit platform to discuss the problem or lack of approach they are facing for a given problem.

We will be making software which will directly enable the interaction of users with interview bit system. The system will be having a home in which all the blogs will be displayed. for a particular blog there will be a description a heading to it. Different users can give the comment on that particular blog and help the other user.

IB Discuss product has aimed to attract a greater number of users to visit its platform and share their experience while looking for doubts and will have the better optimized approach and the constraint which they were looking for a certain problem.

This system can reduce the confusions and will provide better user experience by providing an interactive console it will satisfy the users very well. Our software will be very helpful to the passionate coders which are looking for a meaningful content and solving good amount of challenges with in the time constraint.

### **1.2 Objectives:**

We need to make a change in the way our platform of how users operate. Communicating with a very large community and trying to solve a particular task is a very tedious and time-consuming task so IB Discuss bring such users on their platform to communicate with each

other. While this can be done on platform where owner directly can resolve or view its customer issue.

To avoid all the hard work done by solving the question and getting a better approach is must feature of IB Discuss where a filtered result come and users can communicate to know or grasp the better understanding.

The intended objectives of this project are: -

- To provide the guest user the interaction and the results which they are looking for the query.
- To allow interview bit users to create blog and provide a coding description they are looking for.
- To allow interview bit users to edit the blog as mistakes in writing can take place.
- To allow interview bit users to delete the blog if they find it is of inappropriate value.
- To allow interview bit users to comment on any blog which has been posted so that they can convey their thoughts about the problem.
- To have check on the guest users they are not allowed to comment or create a blog until they are logged in to interview bit.
- Every user can view each and every blog which has been posted by the user as it can help all the users to resolve their and good knowledge about a particular concept.
- To provide passport-based authentication so that users cannot login indirectly from any URL.

To make the website user friendly with clear and understandable instructions.

### **1.3 Significance / Scope:**

IB Discuss provide different users to discuss their doubts or approach on a common platform by which other can get a better understanding. Generally during the coding challenges like Google Codejam, Google Kickstart coders do not get an advantage of finding out the solution of different problems and knowing a better and an optimized way of a challenge and the users come together on a platform to discuss such challenges they face solving challenges.

IB Discuss provides blogs in which user can write problem heading, and a description in that they can either provide a code or challenge which they are facing, they can also mention the problem description or question with the time and space constraints so that other users can identify the better and correct approach.

With the help of IB Discuss the above 6,00,000 of its users can review the blog and give its thoughts about it and also the person which has posted the blog will be getting different approaches and views of the people and the most important thing they can communicate with each other on a public platform with help of comments which help them to configure what other person is trying to say.

## **CHAPTER 2.**

### **SYSTEM REQUIREMENT ANALYSIS**

#### **2.1 Information Gathering**

In IB Discuss there are module which requires information to be collected as to what domain they work or how to provide them better functionalities. As in the main emphasis in the IB Discuss is to bring all its users on a single platform so that can interact with each and solve each other doubts with the help of blogging.

Guest users which will only able to view all the blogs as to grant their attention to one of the products of interview bit that is IB Discuss. Guest users will be able to view all the blogs and also will be able to view all the comments on the blog but not able to participate on the blog and the comments if they try do to so they will be directed to a login page with a flash message.

Interview Bit users will be able to view all the blogs and also create their own blog, edit the blog and delete the blog which a he/she has created. Users will also be allowed to put their views on another blog with the help of comments so that a better understanding is developed.

#### **2.2 System Feasibility**

##### **2.2.1 Economical**

IB Discuss provides effective way to connect multiple users by which they can increase their community and also raise their funds if the IB discuss is used widely so that more users will love to code on the interview bit platform.

With the help of IB Discuss, interview bit can hold various interview on the on their platform so that it can increase their publicity and also enhance its users to land over to get a good job.

### **2.2.2 Technical**

IB Discuss provides an edge over to the interview bit as on the other websites like codechef, codeforces all have their discussions board so to provide our users with better blogging so that easy to find a query a coder is looking being search effective and optimized way so that he can easily look for a worldwide level contests and also would be effective to the new comers who are just started the coding.

IB Discuss uses mongo DB which the help of mongoose API so that it can easily shift to elastic search for a scalable data if needed.

### **2.2.3 Behavioral**

IB Discuss will help the its users to bring doubts on its platform and discuss various challenges they face during solving the solving data structure and algorithm problems generally in contests and problem solving the coders face challenges they generally do not get different people to review their code or review the analysis of complexity and also better and optimized approach.

## **2.3 Platform Specification**

### **2.3.1 Hardware**

IB Discuss can run on any system it must just has a browser and internet access and the users can use either laptops, tablets, mobile phones or PCs it is supported on all the website.

But for the server it requires npm and mongodb to installed on the instance. Mongoose provides edge over the database because it can be non-relational accordingly can easily handle scalable data and for structure we can use elastic search which can provide a better handling of data.

### **2.3.2 Software Implementation Technology**

The project is having been built on a Unix/Linux environment in which it has its libraries imported from npm (Node package manager). The framework used for IB Discuss is Express JS which provides our internal socket and routing management so that it can be hosted on to a

HTTP and also for routing I have used RESTful routes so that easy data is transferred and the costing is also reduced. Backend work is done in node js. For the authentication and authorization purpose of users we have used passport library.

The database used here is mongo DB which is non-relational database we have used its mongoose API which connect with express to easily transfer the data to the database and also for the front end we have used Bootstrap, HTML and CSS. The project has been deployed to Amazon EC2 instance.



## Chapter 3.

# Experimental Setup

### 3.1 Landing Page

The landing page of website provide two options to user

3.1.1 undergo authorization

3.1.2 continue as guest user

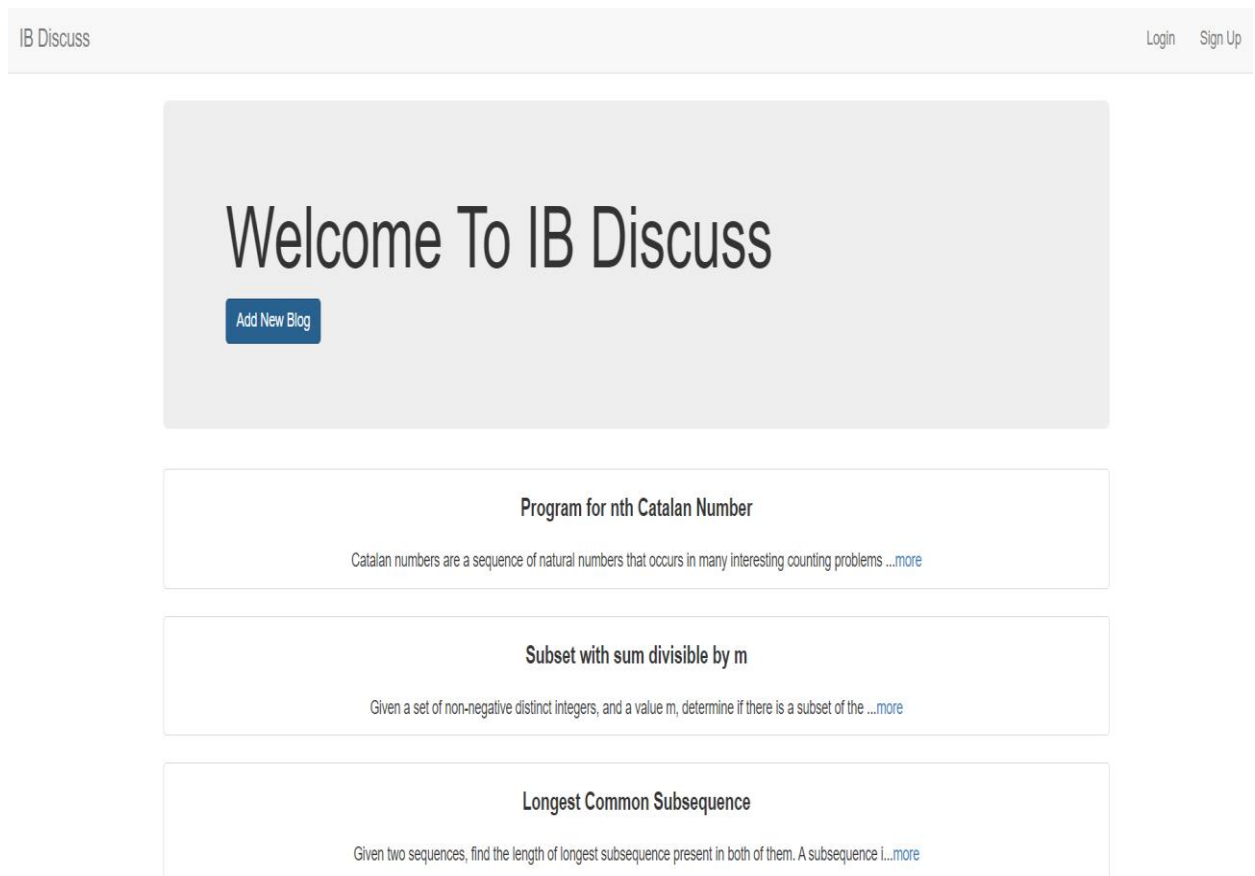


Fig 3.1 Landing page

### 3.2 Authorization of user

As shown in the figure the login page contains two fields ie. Username and password. Also the alert is created using flash library from npm.

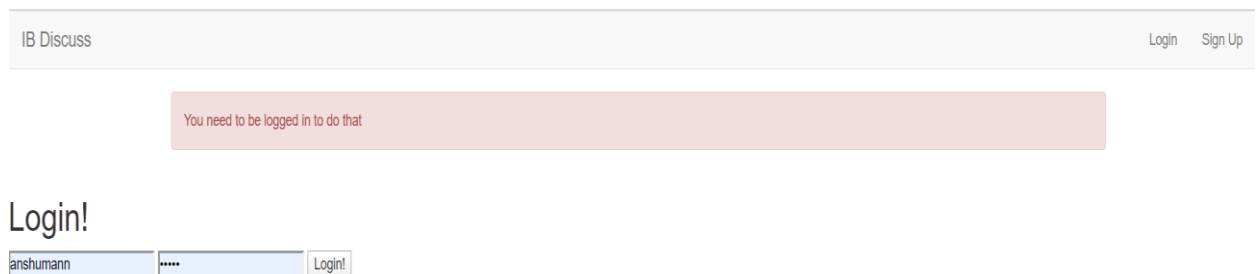


Fig 3.2 Login Page

### 3.3 Post a blog

An authorised user can post a blog, as shown in the figure, he can add a title, question and description related to it. A guest user cannot do so.

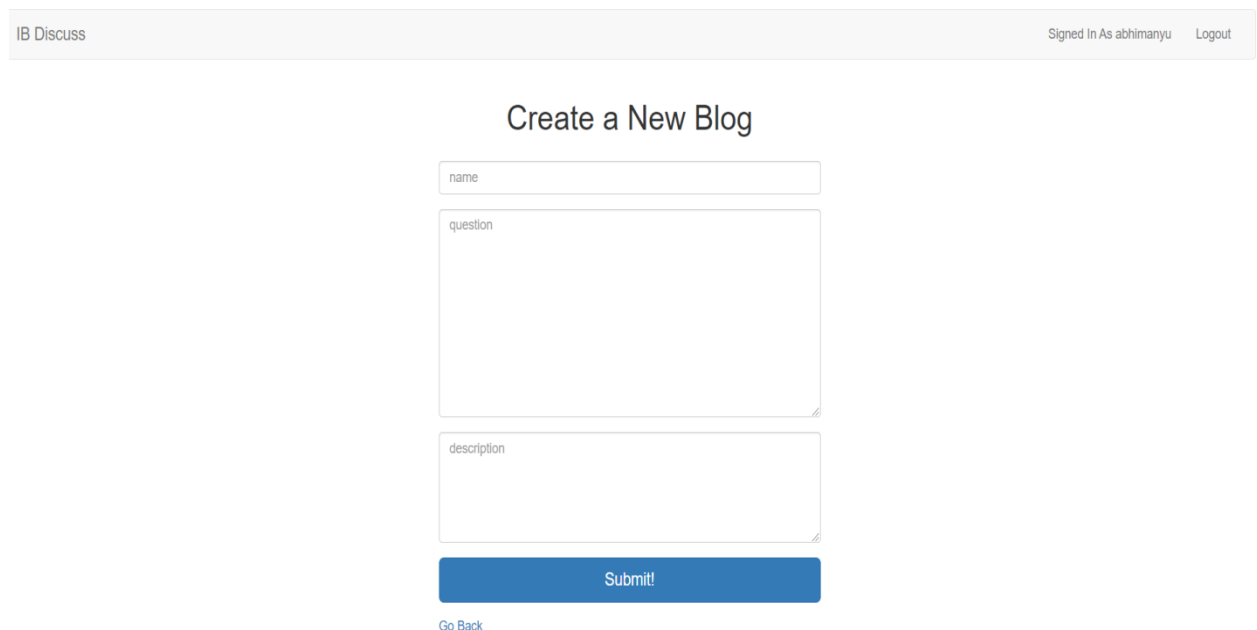


Fig 3.3 New Blog

### 3.4 Editing and deleting a post

The creator of the post can only edit or delete the blog.

Signed In As yusra Logout

## Longest Common Subsequence

Given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example, "abc", "abg", "bdf", "aeg", "acefg", .. etc are subsequences of "abdefg". Examples: LCS for input Sequences "ABCDGH" and "AEDFHR" is "ADH" of length 3. LCS for input Sequences "AGGTAB" and "GXTXAYB" is "GTAB" of length 4.

The naive solution for this problem is to generate all subsequences of both given sequences and find the longest matching subsequence. This solution is exponential in term of time complexity. Let us see how this problem possesses both important properties of a Dynamic Programming (DP) Problem. 1) Optimal Substructure: Let the input sequences be  $X[0..m-1]$  and  $Y[0..n-1]$  of lengths  $m$  and  $n$  respectively. And let  $L(X[0..m-1], Y[0..n-1])$  be the length of LCS of the two sequences  $X$  and  $Y$ . Following is the recursive definition of  $L(X[0..m-1], Y[0..n-1])$ . If last characters of both sequences match (or  $X[m-1] == Y[n-1]$ ) then  $L(X[0..m-1], Y[0..n-1]) = 1 + L(X[0..m-2], Y[0..n-2])$  If last characters of both sequences do not match (or  $X[m-1] != Y[n-1]$ ) then  $L(X[0..m-1], Y[0..n-1]) = \text{MAX} ( L(X[0..m-2], Y[0..n-1]), L(X[0..m-1], Y[0..n-2]) )$  Time complexity of the above naive recursive approach is  $O(2^n)$  in worst case and worst case happens when all characters of  $X$  and  $Y$  mismatch i.e., length of LCS is 0. Considering the above implementation, following is a partial recursion tree for input strings "AXYT" and "AYZX"

Submitted By yusra

Edit Delete

Fig 3.4 Edit/Delete blog

### 3.5 View Blogs and comments

Every user can view blogs and comments.

Signed In As abhimanyu Logout

## Longest Common Subsequence

Given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example, "abc", "abg", "bdf", "aeg", "acefg", .. etc are subsequences of "abdefg". Examples: LCS for input Sequences "ABCDGH" and "AEDFHR" is "ADH" of length 3. LCS for input Sequences "AGGTAB" and "GXTXAYB" is "GTAB" of length 4.

The naive solution for this problem is to generate all subsequences of both given sequences and find the longest matching subsequence. This solution is exponential in term of time complexity. Let us see how this problem possesses both important properties of a Dynamic Programming (DP) Problem. 1) Optimal Substructure: Let the input sequences be  $X[0..m-1]$  and  $Y[0..n-1]$  of lengths  $m$  and  $n$  respectively. And let  $L(X[0..m-1], Y[0..n-1])$  be the length of LCS of the two sequences  $X$  and  $Y$ . Following is the recursive definition of  $L(X[0..m-1], Y[0..n-1])$ . If last characters of both sequences match (or  $X[m-1] == Y[n-1]$ ) then  $L(X[0..m-1], Y[0..n-1]) = 1 + L(X[0..m-2], Y[0..n-2])$  If last characters of both sequences do not match (or  $X[m-1] != Y[n-1]$ ) then  $L(X[0..m-1], Y[0..n-1]) = \text{MAX} ( L(X[0..m-2], Y[0..n-1]), L(X[0..m-1], Y[0..n-2]) )$  Time complexity of the above naive recursive approach is  $O(2^n)$  in worst case and worst case happens when all characters of  $X$  and  $Y$  mismatch i.e., length of LCS is 0. Considering the above implementation, following is a partial recursion tree for input strings "AXYT" and "AYZX"

Submitted By yusra

Add New Comment

naman

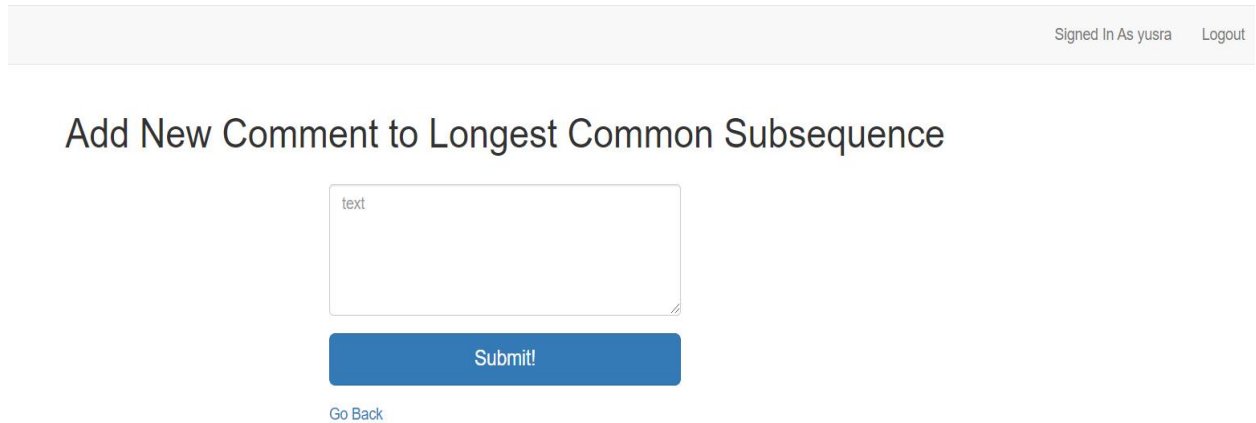
```
def lcs(X, Y, m, n): if m == 0 or n == 0: return 0; elif X[m-1] == Y[n-1]: return 1 + lcs(X, Y, m-1, n-1); else: return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));
```

Fig 3.5 View Blogs and comments

### 3.6 Adding a comment

An authorized user can add a comment. Also he can edit and delete his own comment.

Comments are stored as array of json objects in mongoose.

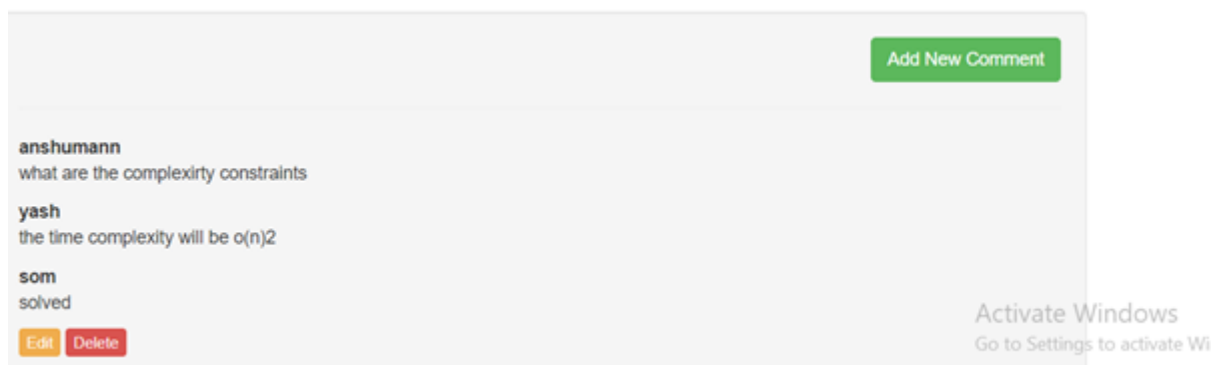


The screenshot shows a web interface for adding a comment. At the top right, there is a navigation bar with the text "Signed In As yusra" and a "Logout" link. Below this, the main heading is "Add New Comment to Longest Common Subsequence". Under the heading is a text input field with the placeholder text "text". Below the input field is a blue button labeled "Submit!". At the bottom left of the form area, there is a link labeled "Go Back".

Fig 3.6 Adding comment

### 3.7 Editing and deleting a comment

The user who had posted the comment can only edit and delete his comment. Flash library from npm is used to create the pop-up.



The screenshot shows a list of comments on a web page. At the top right, there is a green button labeled "Add New Comment". The comments are listed as follows:

- anshumann**  
what are the complexirty constraints
- yash**  
the time complexity will be  $O(n)^2$
- som**  
solved

Below the "solved" comment, there are two buttons: "Edit" (orange) and "Delete" (red). At the bottom right of the page, there is a watermark that says "Activate Windows Go to Settings to activate Wi".

Fig 3.7 Edit/Delete comment

## **Chapter 4.**

### **System Analysis**

#### **4.1 Information Flow Representation**

##### **4.1.1 Activity diagram**

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc. Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements –

- Activities
- Association
- Conditions
- Constraints

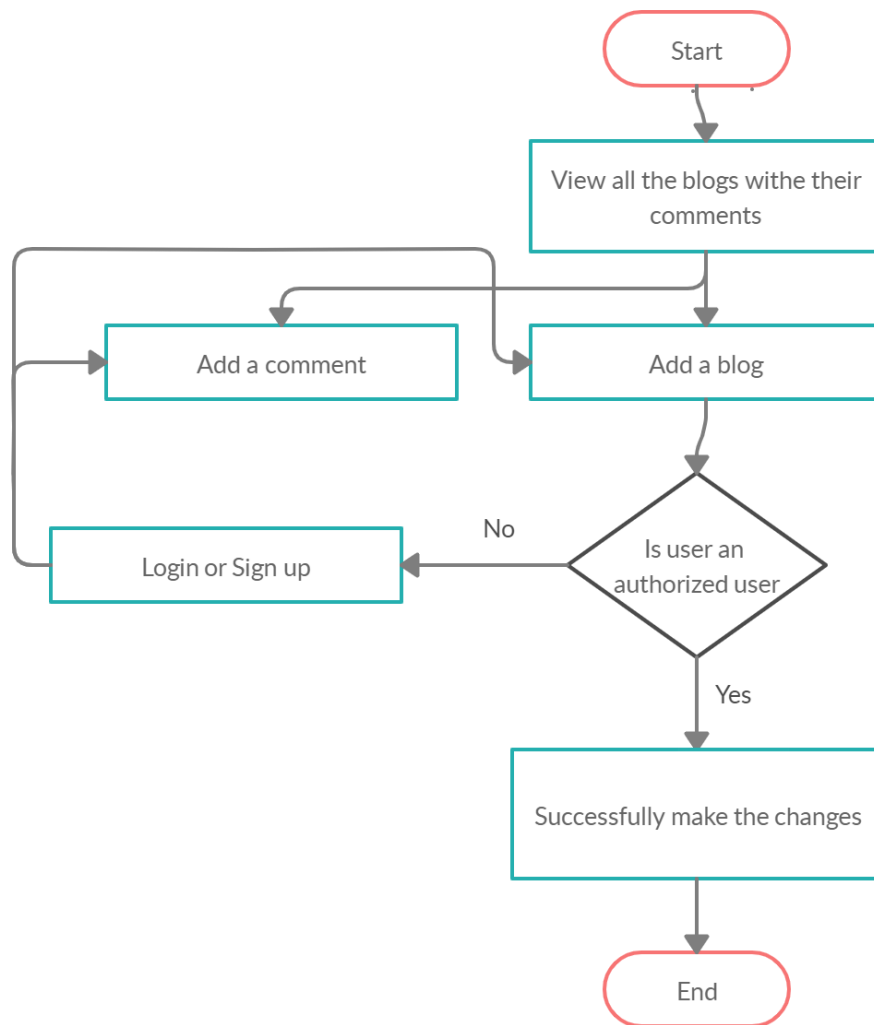


Fig 4.1 Activity Diagram

## 4.2 Use Case Diagrams

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design

requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements and actors.



Fig 4.2 Use-Case Diagram



## 4.3 ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

### 4.3.1 Mapping natural language

ER components can be equated to parts of speech, as Peter Chen did. This shows how an ER Diagram compares to a grammar diagram:

- **Common noun:** Entity type. Example: student.
- **Proper noun:** Entity. Example: Sally Smith.
- **Verb:** Relationship type. Example: Enrolls. (Such as in a course, which would be another entity type.)
- **Adjective:** Attribute for entity. Example: sophomore.
- **Adverb:** Attribute for relationship. Example: digitally.

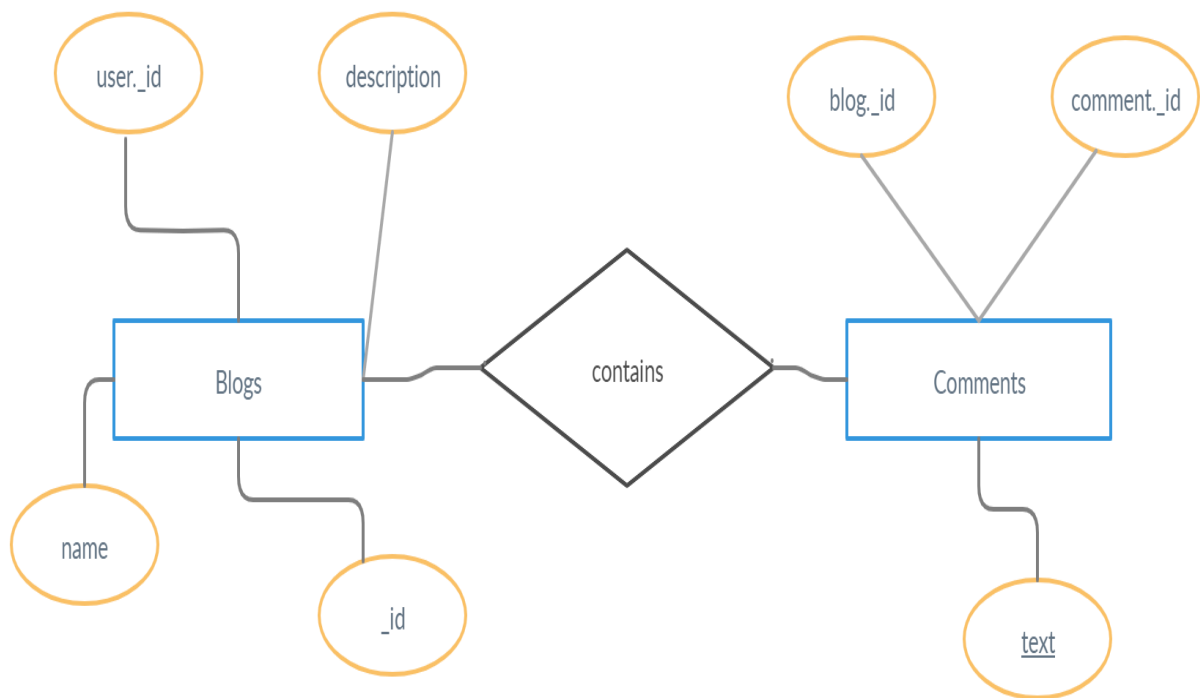


Fig 4.3 ER Diagram

## **CHAPTER 5.**

### **TESTING**

#### **5.1 Testing Objective**

Testing is one of the important phases in the complete development of the project. It allows to check the web application for potential bugs and any other functional errors.

The main objective to test IB Discuss was to check if all the two modules created-Guest module, User module, all are working properly. Their functional working as separate modules as well as the proper flow of all the two modules when they are working together needed to be tested.

#### **5.2 Testing Scope**

We increased the scope of testing by making sure that some additional features worked properly as well. One of this was to test the adding the blog, editing the blog and deleting the blog. The working of creation of blog and the different users commenting on it with three options available to them of create, insert or edit the comment and also to authorize that guest cannot comment or either add a blog until he/she is registered or signed up with interview bit and also the user which has crested the blog can also edit, delete it.

#### **5.3 Testing Methods**

Used Here, manual testing is performed to find out the defects or bugs in a software program. In this method the tester plays an important role of end user and verifies that all the features of the application are working correctly. The tester manually executes test cases without using any automation tools. Test cases are planned to cover almost 100% of the software application. As manual testing involves complete test cases it is a time-consuming test. The differences between

actual and desired results are treated as defects. The defects are then fixed. The tester retests the defects to ensure that defects are fixed. The goal of manual testing is to ensure that application is defect & error free and is working fine to provide good quality work to customers.

## **5.4 Test Cases**

### **1. Test cases for Functionality Testing:**

- There are no broken Outgoing links, Internal links, Anchor Links and Mail to Links.
- Scripting checks on the form are working as expected.
- Checking default values are being populated.
- Checking html and CSS for Syntax Errors, Readable Color Schemas and Standard Compliance.

### **2. Test Cases for Database Testing:**

- Test if any errors are shown while executing queries.
- Data Integrity is maintained while creating, updating or deleting data in database. Test data retrieved from your database is shown accurately in your web application.

### **3. Test Cases for Compatibility Testing:**

- Testing web application on different browsers like Internet Explorer, Firefox, Safari, Opera browsers with different versions.
- Testing web application on different operating systems like Windows, Unix, MAC, Linux, Solaris with different OS flavors.
- Testing web pages on mobile browsers.

### **4. Test cases for Security Testing:**

- Test by pasting the internal URL directly into the browser address bar without login. Internal pages should not open.
- Guest should be only allowed to just view the content they cannot add a comment or create a blog until registered.

## **5. Test cases for Guest Module:**

- The guest should be able to view different blogs posted on the IB Discuss.
- The guest should not be able to put any request for creation or adding a comment in a blog.
- The guest user if tries to create a blog should be redirected to the login page.
- The guest user if tries to add a comment should be redirected to the login page.

## **6. Test cases for User Module:**

- The user should be able to view different blogs posted on the IB Discuss.
- The user should be able to put any request for creation or adding a comment in a blog.
- The user should be able to create a blog, edit his/her blogs, delete his/her blogs.
- The user should be able to comment a blog, edit his/her blogs, delete his/her blogs.

## **5.5 Sample Test Data & Results**

- 1) The website is functionally complete with no bugs identified.
- 2) Database integrity is maintained on any insertion, updating or deletion and the correct data is retrieved whenever a query is run.
- 3) IB Discuss is any browser compatible, operating system compatible and also device compatible.
- 4) Internal pages of the website are not visible without login are also carried out securely. All the modules of the project are working properly having their corresponding limitations as well.

## **CHAPTER 6.**

### **Learning in Training**

#### **6.1 NodeJS**

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

Though .js is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

#### **6.2 ExpressJS**

ExpressJS is a web application framework that provides you with a simple API to build websites, web apps and back ends. With ExpressJS, you need not worry about low level protocols, processes, etc.

Express provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express.

Express was developed by TJ Holowaychuk and is maintained by the Node.js foundation and numerous open source contributors.

Unlike its competitors like Rails and Django, which have an opinionated way of building applications, Express has no "best way" to do something. It is very flexible and pluggable.

Pug (earlier known as Jade) is a terse language for writing HTML templates. It –

- Produces HTML
- Supports dynamic code
- Supports reusability (DRY)

It is one of the most popular template languages used with Express.

## 6.3 NPM

NPM (originally short for Node Package Manager) is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website. The package manager and the registry are managed by npm, Inc.

npm is included as a recommended feature in Node.js installer. npm consists of a command line client that interacts with a remote registry. It allows users to consume and distribute JavaScript modules that are available on the registry. Packages on the registry are in CommonJS format and include a metadata file in JSON format. Over 477,000 packages are available on the main npm registry. The registry has no vetting process for submission, which means that packages found there can be low quality, insecure, or malicious.<sup>[16]</sup> Instead, npm relies on user reports to take down packages if they violate policies by being low quality, insecure or malicious. npm exposes

statistics including number of downloads and number of depending packages to assist developers in judging the quality of packages.

In npm version 6, the audit feature was introduced to help developers identify and fix vulnerability and security issues in installed packages. The source of security issues was taken from reports found on the Node Security Platform (NSP), and has been integrated with npm since npm's acquisition of NSP.

npm can manage packages that are local dependencies of a particular project, as well as globally-installed JavaScript tools. When used as a dependency manager for a local project, npm can install, in one command, all the dependencies of a project through the package.json file.<sup>1</sup> In the package.json file, each dependency can specify a range of valid versions using the semantic versioning scheme, allowing developers to auto-update their packages while at the same time avoiding unwanted breaking changes. npm also provides version-bumping tools for developers to tag their packages with a particular version

## **6.4 Mongoose**

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

MongoDB is a schema-less NoSQL document database. It means you can store JSON documents in it, and the structure of these documents can vary as it is not enforced like SQL databases. This is one of the advantages of using NoSQL as it speeds up application development and reduces the complexity of deployments.

### **6.4.1 Collections**

‘Collections’ in Mongo are equivalent to tables in relational databases. They can hold multiple JSON documents.

### **6.4.2 Documents**

‘Documents’ are equivalent to records or rows of data in SQL. While a SQL row can reference data in other tables, Mongo documents usually combine that in a document.



### **6.4.3Fields**

‘Fields’ or attributes are similar to columns in a SQL table.

### **6.4.4Schema**

While Mongo is schema-less, SQL defines a schema via the table definition. A Mongoose ‘schema’ is a document data structure (or shape of the document) that is enforced via the application layer.

### **6.4.5Models**

‘Models’ are higher-order constructors that take a schema and create an instance of a document equivalent to records in a relational database.

## **6.5 Data Structure and Algorithm**

### **6.5.1 Graph Algorithms**

Topological sorting for Directed Acyclic Graph (DAG) is a linear ordering of vertices such that for every directed edge  $uv$ , vertex  $u$  comes before  $v$  in the ordering. Topological Sorting for a graph is not possible if the graph is not a DAG.

We can modify DFS to find Topological Sorting of a graph. In DFS, we start from a vertex, we first print it and then recursively call DFS for its adjacent vertices. In topological sorting, we use a temporary stack. We don’t print the vertex immediately, we first recursively call topological sorting for all its adjacent vertices, then push it to a stack. Finally, print contents of stack. Note that a vertex is pushed to stack only when all of its adjacent vertices (and their adjacent vertices and so on) are already in stack.

### **6.5.2 Trie**

In computer science, a trie, also called digital tree or prefix tree, is a kind of search tree—an ordered treedata structure used to store a dynamic set or associative array where the keys are usually strings. Unlike a binary search tree, no node in the tree stores the key associated with that node; instead, its position in the tree defines the key with which it is associated; i.e., the value of the key is distributed across the structure. All the descendants of a node have a common prefix of

the string associated with that node, and the root is associated with the empty string. Keys tend to be associated with leaves, though some inner nodes may correspond to keys of interest. Hence, keys are not necessarily associated with every node. For the space-optimized presentation of prefix tree, see compact prefix tree.

In the example shown, keys are listed in the nodes and values below them. Each complete English word has an arbitrary integer value associated with it. A trie can be seen as a tree-shaped deterministic finite automaton. Each finite language is generated by a trie automaton, and each trie can be compressed into a deterministic acyclic finite state automaton.

Though tries can be keyed by character strings, they need not be. The same algorithms can be adapted to serve similar functions on ordered lists of any construct; e.g., permutations on a list of digits or shapes. In particular, a bitwise trie is keyed on the individual bits making up any fixed-length binary datum, such as an integer or memory address.

### **6.5.3 Segment Trees**

In computer science, a segment tree, also known as a statistic tree, is a treedata structure used for storing information about intervals, or segments. It allows querying which of the stored segments contain a given point. It is, in principle, a static structure; that is, it's a structure that cannot be modified once it's built. A similar data structure is the interval tree.

A segment tree for a set  $I$  of  $n$  intervals uses  $O(n \log n)$  storage and can be built in  $O(n \log n)$  time. Segment trees support searching for all the intervals that contain a query point in  $O(\log n + k)$ ,  $k$  being the number of retrieved intervals or segments.

Applications of the segment tree are in the areas of computational geometry, and geographic information systems. The segment tree can be generalized to higher dimension spaces.

## **6.6 Working with Amazon EC2 Instance**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2

to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

### **6.6.1 Features of Amazon EC2**

Amazon EC2 provides the following features:

- Virtual computing environments, known as instances
- Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types
- Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as instance store volumes
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups
- Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses
- Metadata, known as tags, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as virtual private clouds (VPCs)

## **CHAPTER 7.**

### **Summary and Conclusions**

IB Discuss will help the its users to bring doubts on its platform and discuss various challenges they face during solving the solving data structure and algorithm problems generally in contests and problem solving the coders face challenges they generally do not get different people to review their code or review the analysis of complexity and also better and optimized approach.

With the help of IB Discuss the above 6,00,000 of its users can review the blog and give its thoughts about it and also the person which has posted the blog will be getting different approaches and views of the people and the most important thing they can communicate with each other on a public platform with help of comments which help them to configure what other person is trying to say.

During hackathons, challenges, Google Kickstart, Goggle Code Jam, Google Hash Code, Codevita, Infyq and many other big coding competitions the analysis or the discussion board with so many number of users will be provided by the Interview Bit so that it can help its users to communicate with the other thoughts and also help new comers find out how the problem is approached with so many number of users Interview Bit has an edge over other platforms.

## **CHAPTER 8.**

### **Results and Discussion**

IB Discuss will bring more of users to its platform because the product that is developed will help different users and also increase its popularity so that the both users and company will get the advantage of having this product.

Interview Bit team has been planning to set up and launch the products with all the functionality covered and make the system more consistent and scalable so that the blogs which are being posted on the IB Discuss get a better response and users can also take advantage of sharing their experience with IB community so that different other users either guest or IB user does have an advantage over other platforms and can prepare for the interviews solely on one system.

IB Discuss has also provided a good server with the help of horizontal scaling on instances so that its server code can run on different machine while balancing the loads of various instances and also providing a less time response and also an effective blog which users is looking through and this products can get edge over other platforms because other platforms do not provide a good interface designed with users perspective so that he can easily find the problems and write his or her own blogs.

IB Discuss product has aimed to attract a greater number of users to visit its platform and share their experience while looking for doubts and will have the better optimized approach and the constraint which they were looking for a certain problem.

## CHAPTER 9.

### References

- <https://www.npmjs.com/package/express> (Express Framework)
- <https://github.com/Automattic/mongoose> (Mongoose)
- <https://expressjs.com/en/guide/routing.html> (Routing Techniques)
- <https://github.com/jaredhanson/passport-local> (User Authorization)
- <https://explore.easyprojects.net/blog/guest-users> (Guest Users)
- <https://www.aosabook.org/en/distsys.html> (Scalable Systems)
- <https://searchapparchitecture.techtarget.com/definition/RESTful-API> (RESTful-API)
- <https://nodejs.org/en/docs/> (Node JS)
- [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction)(Express-NodeJS)
- [medium.com/@SigniorGratiano/mongoose-and-express-68994cfdeff](https://medium.com/@SigniorGratiano/mongoose-and-express-68994cfdeff)(Express-Mongoos)
- <http://web.stanford.edu/class/archive/cs/cs166/cs166.1146/lectures/09/Small09.pdf>(Trie)
- <https://www.sciencedirect.com/topics/computer-science/topology-graph>(Graph Topology)
- <https://dzone.com/articles/understanding-the-cap-theorem>(CAP theorem)